Chapter 7 Approximating x^2
Error bounds for approximations with deep ReLU networks
Dmitry Yarotsky

Fang Zhiying

CUHK(Shenzhen)

fangzhiying@cuhk.edu.cn

March 11, 2021

Why x^2

- With x^2, polarization gives us multiplication.
- $xy = \frac{1}{2}\left((x+y)^2 - x^2 - y^2\right)$.
- Then we get monomials, polynomials,...

- Define $S_i := \left(\frac{0}{2^i}, \frac{1}{2^i}, \cdots, \frac{2^i}{2^i}\right)$
- Let h_i be the linear interpolation of x^2 on S_i

Backgrounds

- For one hidden layer neural networks, to approximate a C^n function on a d dimensional set with infinitesimal error ϵ one needs a network of size about $\epsilon^{-\frac{d}{n}}$, assuming a smooth activation function [1, 2].
- In this paper, Yarotsky consider L^∞ error of approximation of functions belonging to the Sobolev space $W_\infty^n\left([0,1]^d\right)$. We use $F_{d,n}$ to denote the unit ball in $W_\infty^n\left([0,1]^d\right)$. The norm in $W_\infty^n\left([0,1]^d\right)$ can be defined by

$$\|f\|_{W_\infty^n([0,1]^d)} = \max_{\mathbf{n}:|\mathbf{n}|\leq n} \operatorname{ess} \sup_{\mathbf{x}\in[0,1]^d} \left|D^{\mathbf{n}} f(\mathbf{x})\right|,$$

where $\mathbf{n} = (n_1, \cdots, n_d) \in \{0, 1, \cdots\}^d$, and $|\mathbf{n}| = n_1 + \cdots + n_d$.

Results

Proposition 1

Let $\rho : \mathbb{R} \to \mathbb{R}$ be any continuous piece wise linear function with M breakpoints, where $1 \leq M < \infty$.

A *Let ξ be a network with the activation function ρ, having depth L, W weights and U computation units. Then there exists a ReLU network η that has depth L, not more than $(M+1)^2 W$ weights and not more than $(M+1)U$ units, and that computes the same function as ξ.*

B *Conversely, let η be a ReLU network of depth L, W weights and U computation units. Let $\mathcal{D}$ be a bounded subset of $\mathbb{R}^d$.Then there exists a network with activation function ρ that has depth L, $4W$ weights and not more than $2U$ units, and that computes the same function as η on the set $\mathcal{D}$.*

Idea

- Part A of Proposition 1 can be easily done by noticing that

$$\rho(x) = c_0\sigma(a_1 - x) + \sum_{m=1}^{M} c_m\sigma(x - a_m) + h$$

where $a_1 < \cdots < a_M$ be the breakpoints of ρ. It means that the computations of a ρ-unit can be equivalently represented by a linear combinations of a constant function and $M + 1$ σ units.

Idea

- For part B of proposition 1, let a be any breaking point of ρ such that $\rho'(a+) \neq \rho'(a-)$. Let r_0 be the distance separating a from the nearest other breaking point, so that ρ is linear on $[a, a + r_0]$ and on $[a - r_0, a]$. Then for any $r > 0$, we have

$$\sigma(x) = \frac{\rho(a + \frac{r_0 x}{2r}) - \rho(a - \frac{r_0}{2} + \frac{r_0 x}{2r}) - \rho(a) + \rho(a - \frac{r_0}{2})}{(\rho'(a+) - \rho'(a-))\frac{r_0}{2r}},$$

for any $x \in [-r, r]$. We only need to take r large enough to get the result.

Results

Proposition 2

The function $f(x) = x^2$ *on the segment* $[0, 1]$ *can be approximated with any error* $\epsilon > 0$ *by a ReLU network having the depth and the number of weights and computation units* $O(\ln(\frac{1}{\epsilon}))$.

Idea

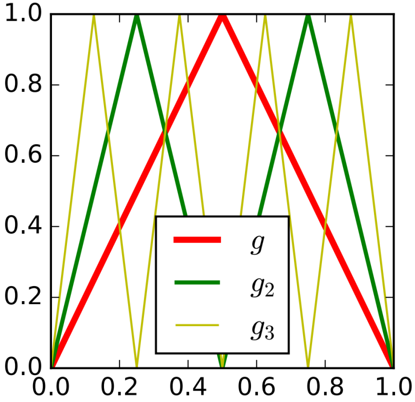
Consider the "tooth" function $g : [0,1] \to [0,1]$,

$$g(x) = \begin{cases} 2x, & x < \frac{1}{2}, \\ 2(1-x), & x \geq \frac{1}{2}, \end{cases}$$

and the iterated functions $g_s(x) = \overbrace{g \circ g \circ \cdots \circ g}^{s}(x)$.

$$g_s(x) = \begin{cases} 2^s \left(x - \frac{2k}{2^s}\right), & x \in \left[\frac{2k}{2^s}, \frac{2k+1}{2^s}\right], k = 0, 1, \cdots, 2^{s-1} - 1 \\ 2^s \left(\frac{2k}{2^s} - x\right), & x \in \left[\frac{2k-1}{2^s}, \frac{2k}{2^s}\right], k = 1, 2, \cdots, 2^{s-1}. \end{cases}$$

Tooth function

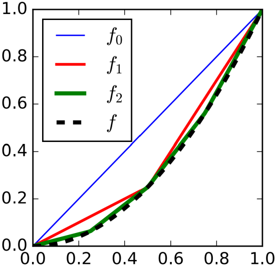


(a)

Linear interpolation

Let f_m be the piece wise linear interpolation of f with $2^m + 1$ uniformly distributed breakpoints $\frac{k}{2^m}, k = 0, \cdots, 2^m$.

$$f_m\left(\frac{k}{2^m}\right) = \left(\frac{k}{2^m}\right)^2, \ k = 0, \cdots, 2^m.$$



(b)

How to estimate x^2

- The function f_m approximates $f = x^2$ with the error $\epsilon_m = 2^{-2m-2}$.
- $$f_{m-1}(x) - f_m(x) = \frac{g_m(x)}{2^{2m}}. \tag{1}$$
- $$f_m(x) = x - \sum_{s=1}^{m} \frac{g_s(x)}{2^{2s}}. \tag{2}$$
- $$g(x) = 2\sigma(x) - 4\sigma(x - \frac{1}{2}) + 2\sigma(x - 1). \tag{3}$$

Results

$$xy = \frac{1}{2}\left((x+y)^2 - x^2 - y^2\right).$$

Proposition 3

Given $M > 0$ and $\epsilon \in (0,1)$, there is a ReLU network η with two input units that implements a function $\tilde{\times} : \mathbb{R}^2 \to \mathbb{R}$ so that

- *for any inputs x, y, if $|x| \leq M$ and $|y| \leq M$, then $\left|\tilde{\times}(x,y) - xy\right| \leq \epsilon$;*
- *if $x = 0$ or $y = 0$, then $\tilde{\times}(x,y) = 0$;*
- *the depth and the number of weights and computation units in η are not great than $c_1 \ln \frac{1}{\epsilon} + c_2(M)$.*

What we have

Let $\tilde{f}$ be the approximate square function from Proposition 2 such that $\tilde{f}(0) = 0$ and $\left|\tilde{f}(x) - x^2\right| < \delta$ for $x \in [0, 1]$.
Set

$$\tilde{\times}(x, y) = \frac{M^2}{8}\left(\tilde{f}(\frac{|x+y|}{2M}) - \tilde{f}(\frac{|x|}{2M}) - \tilde{f}(\frac{|y|}{2M})\right)$$

with $\delta = \frac{8\epsilon}{3M^2}$.

- Product Gate;
- Polynomial.

$$xy = \frac{1}{2}\left((x+y)^2 - x^2 - y^2\right).$$

Main results

Theorem 1

For any d, n and $\epsilon \in (0,1)$, there is a ReLU network such that

1. *is capable of expressing any function from $F_{d,n}$ with error ϵ;*
2. *has the depth at most $c(d,n)\left(\ln\frac{1}{\epsilon}+1\right)$ and at most $c(d,n)\epsilon^{-\frac{d}{n}}\left(\ln\frac{1}{\epsilon}+1\right)$ weights and computation units.*

Idea

1. Approximate function f by a sum-product combination f_1 of local Taylor polynomials and one-dimensional piecewise-linear functions.
2. Use results of previous Propositions to approximate f_1 by a neural network.

Idea

Let N be a positive integer. Consider a partition of unity formed by a grid of $(N+1)^d$ function $\phi_{\mathbf{m}}$ on the domain $[0,1]^d$

$$\sum_{\mathbf{m}} \phi_{\mathbf{m}}(\mathbf{x}) = 1, \quad \mathbf{x} \in [0,1]^d$$

Here $\mathbf{m} = (m_1, \cdots, m_d) \in \{0, 1, \cdots, N\}^d$ and the function $\phi_{\mathbf{m}}$ is defined as

$$\phi_{\mathbf{m}}(\mathbf{x}) = \Pi_{k=1}^{d} \psi \left(3N \left(x_k - \frac{m_k}{N} \right) \right)$$

Idea

$$\psi(x) = \begin{cases} 1, & |x| < 1, \\ 0, & |x| > 2, \\ 2 - |x|, & 1 \le |x| \le 2. \end{cases}$$

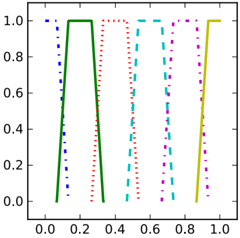


Fig. 3. Functions $(\phi_m)_{m=0}^{5}$ forming a partition of unity for $d = 1, N = 5$ in the proof of Theorem 1.

For any $\mathbf{m} = (m_1, \cdots, m_d) \in \{0, 1, \cdots, N\}^d$, consider the degree $n - 1$ Taylor polynomial for the function f at $\mathbf{x} = \frac{\mathbf{m}}{N}$

$$P_{\mathbf{m}}(\mathbf{x}) = \sum_{\mathbf{n}:|\mathbf{n}|<n} \frac{D^{\mathbf{n}} f}{\mathbf{n}!}\Big|_{\mathbf{x}=\frac{\mathbf{m}}{N}} \left(\mathbf{x} - \frac{\mathbf{m}}{N}\right)^{\mathbf{n}}$$

with $\mathbf{n}! = \Pi_{k=1}^{d} n_k!$ and $\left(\mathbf{x} - \frac{\mathbf{m}}{N}\right)^{\mathbf{n}} = \Pi_{k=1}^{d} \left(x_k - \frac{m_k}{N}\right)^{n_k}$

Taylor polinomial

Now we define $f_1 = \sum_{\mathbf{m} \in \{0,\cdots,N\}^d} \phi_{\mathbf{m}} P_{\mathbf{m}}$. By some calculations we have

$$|f(\mathbf{x}) - f_1(\mathbf{x})| \leq \frac{2^d d^n}{n!} \left(\frac{1}{N}\right)^n .$$

If we choose $N = \left\lceil \left(\frac{n!}{2^d d^n} \frac{\epsilon}{2}\right)^{-\frac{1}{n}} \right\rceil$, then we have

$$\|f - f_1\|_\infty \leq \frac{\epsilon}{2}.$$

Taylor polinomial

$$P_{\mathbf{m}}(\mathbf{x}) = \sum_{\mathbf{n}:|\mathbf{n}|<n} a_{\mathbf{m},\mathbf{n}} \left(\mathbf{x} - \frac{\mathbf{m}}{N}\right)^{\mathbf{n}}$$

$$f_1 = \sum_{\mathbf{m}\in\{0,\cdots,N\}^d} \sum_{\mathbf{n}:|\mathbf{n}|<n} a_{\mathbf{m},\mathbf{n}} \phi_{\mathbf{m}}(\mathbf{x}) \left(\mathbf{x} - \frac{\mathbf{m}}{N}\right)^{\mathbf{n}}.$$

We can see that this expansion is a linear combination of not more than $n^d (N+1)^d$ terms $\phi_{\mathbf{m}}(\mathbf{x}) \left(\mathbf{x} - \frac{\mathbf{m}}{N}\right)^{\mathbf{n}}$ and each of these terms is a product of at most $d+n-1$ piece wise linear univariate factors: d functions $\psi(3Nx_k - 3m_k)$ and at most $n-1$ linear expressions $x_k - \frac{m_k}{N}$.

Taylor polinomial

Now we can implement an approximation of this product by a neural network by using Proposition 3 for $M = d + n$ and some accuracy δ.
The approximation of the product can be $\phi_{\mathbf{m}}(\mathbf{x})\left(\mathbf{x}-\frac{\mathbf{m}}{N}\right)^{\mathbf{n}}$ obtained by the chained application of $\tilde{\times}$ as

$$\tilde{f}_{\mathbf{m},\mathbf{n}}(\mathbf{x}) = \tilde{\times}\left(\psi(3Nx_1 - 3m_1), \tilde{\times}\left(\psi(3Nx_2 - 3m_2), \cdots, \right)\right)$$

By Propsition 3 then we know that $\tilde{f}_{\mathbf{m},\mathbf{n}}(\mathbf{x})$ can be implemented by a ReLU network with the depth and the number of weights and computation units not larger than $(d+n)c(d,n)\ln\frac{1}{\delta}$.

Taylor polinomial

Now we can construct our approximation of f_1 by a neural network output function $\tilde{f}$ as

$$\tilde{f} = \sum_{\mathbf{m} \in \{0, \cdots, N\}^d} \sum_{\mathbf{n} : |\mathbf{n}| < n} a_{\mathbf{m},\mathbf{n}} \tilde{f}_{\mathbf{m},\mathbf{n}}$$

By some careful analysis we get

$$\left| \tilde{f} - f_1 \right| \leq 2^d d^n (d + n) \delta$$

Now we set $\delta = \frac{\epsilon}{2^{d+1} d^n (d+n)}$, then we have $\left\| \tilde{f} - f_1 \right\|_\infty \leq \epsilon$ and hence

$$\left\| f - \tilde{f} \right\|_\infty \leq \epsilon$$

Main results

Theorem 2

For any function $f \in F_{1,1}$ and $\epsilon \in (0, \frac{1}{2})$, there exists a depth 6 ReLU network (with architecture depending on f) that provides an ϵ approximation of f while having not more than $\frac{c}{\epsilon \ln \frac{1}{\epsilon}}$ weights, connections and computation units.

For function $f \in F_{d,n}$

$$\epsilon^{-\frac{d}{n}} \left(\ln \frac{1}{\epsilon} + 1 \right).$$

Lower bounds

Theorem 3

Fix d and n.

- *For any $\epsilon \in (0,1)$, a ReLU network architecture capable of approximating any function $f \in F_{d,n}$ with error ϵ must have at least $c_{d,n}\epsilon^{\frac{-d}{2n}}$.*
- *Let $p \geq 0$, $c_1 > 0$ be some constants. For any $\epsilon \in (0, \frac{1}{2})$, if a ReLU network architecture of depth $L \leq c_1 \ln^p(\frac{1}{\epsilon})$ is capable of approximating any function $f \in F_{d,n}$, then the network must have at least $c_{d,n,p,c_1}\epsilon^{\frac{-d}{n}} \ln^{-2p-1}(\frac{1}{\epsilon})$.*

For function $f \in F_{d,n}$ upper bounds, the depth is at most $c(d,n)\left(\ln\frac{1}{\epsilon}+1\right)$ and the computation units is at most $\epsilon^{-\frac{d}{n}}\left(\ln\frac{1}{\epsilon}+1\right)$.

Advantage of deep nets

Theorem 4

Let $f \in C^2\left([0,1]^d\right)$ be a nonlinear function (i.e not of the form $f(x_1, \cdots, x_d) = a_0 + \sum_{k=1}^{d} a_k x_k$ on the whole $[0,1]^d$). Then, for any fixed L, a depth L ReLU network approximating f with error $\epsilon \in (0,1)$ must have at least $c\epsilon^{\frac{-1}{2(L-2)}}$ weights and computation units, with some constant $c = c(f, L) > 0$.

For function $f \in F_{d,n}$ upper bounds, the computation units is at most $\epsilon^{-\frac{d}{n}}\left(\ln \frac{1}{\epsilon} + 1\right)$. If $\frac{d}{n} < \frac{1}{2(L-2)}$ and $n > 2$ such that $W^{n,\infty}([0,1]^d) \subset C^2([0,1]^d)$.

Thank You!

References I

Hrushikesh Narhar Mhaskar.
Approximation properties of a multilayered feedforward artificial neural network.
Advances in Computational Mathematics, 1(1):61–80, 1993.

Allan Pinkus.
Approximation theory of the mlp model in neural networks.
Acta numerica, 8(1):143–195, 1999.

The End